Auto Fill Fields for ACT!2005/2006

Installation and Usage

Auto Fill Fields automatically loads any number of ACT! fields with values determined by the contents of some other field.

Now you can speed up your data entry tasks by setting up **Auto Fill Fields** specifications for any number of fields in your ACT! databases. Tell **Auto Fill Fields** which field holds the "independent" value (such as the Zip Code) and which fields depend on that value (such as City and State). Then provide the series of determining and dependent values (such as 94105/San Francisco/CA).

You can reference fields from the "current" Contact record, the "current" Group record and the "current" Company record in the same **Auto Fill Fields** specification, making it easy to update company and group information based on contact information, and vice versa.

Installation

1. Shut down all running programs
2. Run **`Install_AutoFillFields.exe`**
3. Follow the instructions on the screen. Restart your computer if prompted to do so.

This will create (if necessary) the `\TNHG` subfolder within the folder that Windows "knows" as the "Program Files" folder – usually `c:\Program Files`. It will create the `\TNHG\AutoFillFields` subfolder, and install **`AutoFillFieldsTrigger.exe`**, and other supporting files, into that folder. The installation process will also copy into the `\ACT\Act for Win 7\Plugins` and into the `\ACT\Act for Windows\Plugins` subfoldesr (within that same "Program Files" folder) the supporting files necessary to operate within ACT!2005/2006.

Usage

The command to execute **Auto Fill Fields** consists of the entire path-and-filename of the Auto Fill Fields program, followed by a "command' or "argument string". ". Please note that if there are any "blanks" in the path-and-filename, then **it must be enclosed in "quotes"** – Windows treats the blank as a "terminator", and the quotes are necessary to tell Windows that the string continues past the blank.

The Program Name

If the path-and-filename contains any blanks (such as `c:\program files\tnhg…`), then that entire path-and-filename must be enclosed in quotes, such as:

```
"C:\program files\tnhg\AutoFillFields\AutoFillFieldsTrigger.exe"
```

The Command String

The "command string" consists of entries that start with a "question mark", followed by a single letter, and "equal sign" and then more information. For example:

```
?V=T

?F=S:\Triggers\SetTheseFields.txt
```

Each "Question mark-Letter-Equals" string is used to tell Auto Fill Fields a specific detail of its operation:

?V= indicates whether or not you want to view the Auto Fill Fields window while it is performing its copy operation.

- `?V=T` indicates that you DO want to see the window, and that you will need to click the [OK] button on that window when the copying is complete.
- `?V=F` indicates that you do NOT want to see the window. This is the default value, and if you don't want to see the window, you can leave out this entire `?V=` "argument"

? F= indicates the name of the text file that contains the information necessary for Auto Fill Fields to do its job. When you use the ?F= argument, the entire-path-and-filename follows the equal-sign. So, for example, you might use

```
?F=S:\ACT\Common\Triggers\FillCityAndState.txt
```

if your specification is in the `FillCityAndState.txt` file, in the subfolder `\ACT\Common\Triggers` in the shared folder `S:`.

The Specification File

The specifications are provided in 3 parts: The "Separator", the "Field List", and the "Value Table".

The Separator tells Auto Fill Fields what singe character will be used in the Field List and in the Value Table to separate the entries in the list and table. It can be a "visible", printable character such as ? or / or – or = or any other character that does not appear in any fieldname or in any independent or dependent value. It can also be an "invisible", non-printable character, such as {tab} (a visible character is suggested; the {tab} capability is provided primarily so that you can set up you specifications using Excel, and save the spreadsheet as a tab-delimited file – see the sample provided with the installation).

The Field List tells Auto Fill Fields the **Independent Field** whose value determines what should be loaded into the **Dependent Fields**. The FIRST field specified indicates the INdependent field, and all subsequent fields indicate the DEpendent fields. Each field "spec" is enclosed within "carats" (< >). And, each field "spec" indicates both the "entity" and the field within the entity. For example, <C:User 1> indicates the Contact entity (indicated by the "C" that precedes the colon) field User 1(indicated by the "User 1" that comes after the colon). "C" indicates the CONTACT entity, "G" indicates the GROUP entity and "A" indicates the COMPANY entity. Some more examples: <A:Zip Code> indicates the "Zip Code" field on the Company record; <G:Group Description> indicates the "Group Description" field on the Group record. You can determine the "field name" and what is available by using the Tools menu > Define Fields option and selecting the "type"/entity of interest, and checking the list of fields that you can use to select a field. These are the "display names" that Auto Fill Fields uses to determine which fields are to be used to determine what gets put, where. A complete example is:

```
<A:Zip Code>?<C:Sales Rep>?<C:Sales Office>?<A:Sales Manager>
```

indicates that the Zip Code on the Company record (<A:Zip Code>) should be used to find an entry in the following value table, and when a match is found, load the Contact record Sales Rep field (<C:Sales Rep>) and Sales Office

(C:Sales Office>) fields, and the Company record Sales Manager field (<A:Sales Manager>), with the specified values.

The Value Table shows the "sets" of INdependent and related DEpendent values to be used in deciding what to "fill", and when. The table uses the "separator character" determined by the "Separator" character (see above) to separate the values. Each line is used to determine the Independent value and the corresponding dependent values. For example, a line that reads:

```
94040?Donald Duck?Anaheim?Mickey Mouse
```

could be use to indicate that when the Zip Code is "94040", then the Sales Rep should be set to "Donald Duck", the Sales Office should be set to "Anaheim" and the Sales Manager should be set to "Mickey Mouse". The **first** value, that comes BEFORE the first instance of the "separator" is the INdependent value to be matched to the value in the INdependent field – the first field in the "Field List" (see above) – in this example, "when the Company Zip Code = 94040". Each subsequent value (as determined by the "separator") will be used to "auto fill" the each subsequent field in the "Field List" - in this example, "load the Contact Sales Rep with 'Donald Duck', load the Contact Sales Office with 'Anaheim', and load the Company Sales Manager with 'Mickey Mouse'".

Example
One example of an entire "specification file" is:

```
+
<C:Credit Rating>+<C:Credit Limit>+<C:Collection Agency>
Great+$1,000,000+None
Very Good+$5,000+Joe
Good+$100+Bubba
Poor+$10+The Collection Guys
Terrible+$0+FBI
```

The "+" is designated as the separator.

The Contact "Credit Rating" field is the INdependent field, whose value will be used to search the following table for a "match". The Contact "Credit Limit" and Contact "Collection Agency" will be "Auto Filled" when a match is found.

When the "Credit Rating" is "Great", then the "Credit Limit" will be set to "$1,000,000" and the "Collection Agency" will be set to "None". When the "Credit Rating" is "Very Good", then the "Credit Limit" will be set to "$5,000" and the "Collection Agency" will be set to "Joe". And so on.

See the samples provided as part of the installation process for more examples.

The Batch File

The version of ACT! in commercial release at the time of this writing does not allow a trigger to include a "command string" or "argument list", like the one that tells Auto Fill Fields how to operate. Therefore, the trigger definition must be placed in a "DOS `.bat` file", and that file used as the trigger definition on the ACT! field. When the .bat file "runs", a DOS window will appear on the screen, showing the commands included in the .bat file and waiting for the commands to finish. This can be somewhat disconcerting to users, so you might want to include a command to hide this window. (A utility is provided in the `\TNHG` folder, in case you need one. Its name is `ConsoleTool.exe`, and it uses the `/hide` "switch" to tell DOS to hide the window. An example of its use is included below.)

As an example, you might use Notepad (or any other text editor) to create a file that contains the following lines:

```
"C:\program files\TNHG\ConsoleTool.exe" /hide
"C:\program files\TNHG\AutoFillFields\AutoFillFieldsTrig_
.exe" ?F=C:\Program Files\TNHG\AutoFillFields\FruitsAndVeg_
etables.txt ?V=T
```

(There are really only two lines in this example but the page isn't wide enough to show the entire second line on a line by itself. The underscores _ are not really part of the line – they are there only to indicate that the REAL line continues on the next "text line".)

The above example uses the ConsoleTool to hide the DOS window (/hide), then runs the Auto Fill Fields program (AutoFillFields.exe), shows the window while it operates (?V=T), and uses the `c:\Program Files\TNHG\AutoFillFields\FruitsAndVegetables.txt` file to provide the separator, field list and value table.

If you save this file as

`FruitsAndVegetables.bat`

In the `s:\act\common` folder, then your trigger would be

`s:\act\common\FruitsAndBegetables.bat`

Additional samples are provided with the installation package – check the \TNHG\AutoFillFields folder for other .bat file examples.